DETAILED ACTION

1. Claims 1, 3, 6-7, 11-19 and 21 are presented for examination.

Double Patenting

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 1, 3, 6-7, 11-19 and 21 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-5, 12-13, 19-22, 26-27, and 33-34 of copending Application No. 10/750,589. Although the conflicting claims are not identical, they are not patentably distinct from each other.

4. For example, claim 1 of copending Application No. 10/750,589 recites placing a thread in an inactive state in response to a predetermined condition and sending a message from a semaphore to change the state of the thread. Claim 11 of the present

application performs the substantially the same steps. Claim 1 of copending Application No. 10/750,589 differs only in that the threads are intended to be used to process graphical elements of an image. It would have been obvious to one of ordinary skill to try to extend the teachings to imagine processing.

5. This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

7. Claims 1, 3, 6-7, 11-19 and 21 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

8. Claim 1 lines 6-7 recite “releasing the thread, in response to the thread finishing with the resource, only if the thread is no dependent on any subsequent thread". A careful reading of the specification does not provide any support for this claim limitation. The specification only provides for the releasing of semaphore in paragraphs [0042-0044]. However, releasing semaphores is not the same as releasing threads and no

mention by the specification is made indicating such a step is dependent upon thread dependency. Applicant asserts that paragraph [0024] of the original specification provide support the newly added limitation. However, no mention is made of releasing a thread based on dependency of a subsequent thread is found by the Examiner.

9. Independent claims 11 and 15 are rejected for the same reasons as claim 1 above.

10. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

11. Claims 1, 3, 6-7, 11-19 and 21 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

a. The following terms lack antecedent basis in the claims:

i. Claim 11 line 10, “said execution entity”.

ii. Claim 15 lines 7-8, “said execution entity”.

b. The following terms are not clearly understood in the claims:

iii. Claim 1 line 6 recites, "releasing the thread". It is unclear what is meant by releasing the thread since there is no indication in the previous limitations that a thread has been locked in order to be released. It is also unclear how the releasing step is to be performed (i.e. is the thread being put back in the inactive state?). Lines 6-7 recite, “only if the thread is not dependent on any subsequent thread”. It is unclear how a thread is

dependent on another subsequent thread when the preceding limitations only reference resource dependency. It is also unclear what is mean by a subsequent thread (i.e. how is a subsequent thread determined?).

iv. Claim 11 lines 10-11 recites, "said execution entity to release the thread". It is unclear what is meant by releasing the thread since there is no indication in the previous limitations that a thread has been locked in order to be released. It is also unclear how the releasing step is to be performed (i.e. is the thread being put back in the inactive state?). Lines 11-12 recite, “only if the thread is not dependent on any subsequent thread”. It is unclear how a thread is dependent on another subsequent thread when the preceding limitations only reference resource dependency. It is also unclear what is mean by a subsequent thread (i.e. how is a subsequent thread determined?).

v. Claim 15 lines 7-8 recites, "said execution entity to release the thread". It is unclear what is meant by releasing the thread since there is no indication in the previous limitations that a thread has been locked in order to be released. It is also unclear how the releasing step is to be performed (i.e. is the thread being put back in the inactive state?). Lines 8-9 recite, “only if the thread is not dependent on any subsequent thread”. It is unclear how a thread is dependent on another subsequent thread when the preceding limitations only reference resource dependency. It is also

unclear what is mean by a subsequent thread (i.e. how is a subsequent thread determined?).

Claim Rejections - 35 USC § 103

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

13. Claims 1, 3, 6-7 and 15-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joffe et al. (US Pat No. 6,978,330) in view of Haghighat et al. (US PG Pub No. US 2005/0080760 A1).

14. Regarding claim 1, Joffe teaches a method comprising:

placing an executable thread of instructions in an inactive state in response to a resource being unavailable (col 2 lines 54-58, wherein Task1, Task2, Task3 are suspended while Task0 has been granted the semaphore in order to access a memory location); and

when the resource becomes available, changing the thread of instructions to the active state and granting the resource to the thread of instructions (col 2 lines 51-65, wherein a task is awakened from sleep and wherein tasks must be granted access to semaphore in order to access the memory location).

15. Joffe does not teach releasing the thread, in response to the thread finishing with the resource, only if the thread is not dependent on any subsequent thread. Haghighat teaches that suspending a thread (i.e. releasing the thread) that owns a lock required by a subsequent thread, can result in a deadlock situation if the resumption of the suspended thread relies on the completion of the subsequent thread whose progress is dependent on the lock of the suspended thread ([0004]). Therefore, it would have been obvious to one of ordinary skill in the art to release the thread in response to the thread finishing with the resource only if there is no thread dependency. One would be motivated by the desire to avoid the deadlock situation taught by Haghighat.

16. Regarding claim 3, Joffe teaches executing the thread of instructions when in the active state (col 2 lines 51-65, wherein Task0 must be in an active state in order to execute instructions).

17. Regarding claim 6, Joffe teaches maintaining an indication of a state of each of a plurality of executable threads of instructions (col 2 lines 51-65, wherein it is inherent that Joffe maintains an indication on the state of each thread in order to change them from the wait state).

18. Regarding claim 7, Joffe and Haghighat do not teach that the indication of the state of each thread comprises a state variable corresponding to a dependency, if any, of an associated thread.

19. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Joffe to include a state variable corresponding to a dependency of an associated thread. Joffe teaches that each thread is dependent on the use of the semaphore/resource for operation. Joffe also teaches the need to synchronize communications among threads (col 2 lines 40-44). One would be motivated by the desire to prohibit out of order accesses to resources that could cause system conflicts.

20. Regarding claim 15, Joffe teaches a system comprising:

a memory controller (col 2 lines 40-44, wherein it is inherent a memory controller exists to access shared memory); and

an execution circuit coupled with the memory controller to receive and execute a thread of instructions (col 2 lines 44-58, wherein tasks are executed by a microcontroller), wherein the execution circuit transmits a request message (col 2 lines 51-53, wherein tasks make requests to semaphores) and places the thread in an inactive state in response to the thread of instructions requiring a resource that is unavailable (col 2 lines 54-58, wherein tasks are suspended when a semaphore is held by another thread), said execution unit to automatically change the thread of instructions to an active state (col 2 lines 58-60, wherein the task is awakened from sleep in order to access the resource) and grant the resource to the thread of instructions when the resource becomes available (col 2 lines 51-53, wherein tasks are granted the semaphore).

21. Regarding claim 16, Joffe and Haghighat do not teach further comprising: at least one additional execution circuit to execute threads of instructions; and a thread dispatcher coupled with the execution circuit and at least one additional execution circuit to dispatch threads for execution by selected execution circuits.

22. It would have been obvious to one of ordinary skill in the art, at the time of the invention to add one additional execution circuit to execute threads of instructions and a thread dispatcher. It is well known in the art to add additional execution units to increase processing capability of processors.

23. Regarding claim 17, Joffe and Haghighat do not teach that the execution circuitry, in response to receiving the semaphore acknowledge message, resumes execution of the thread of instructions including accessing the resource associated with the semaphore.

24. Joffe does teach that once the semaphore is granted to another thread, the thread is awakened from sleep and can access the resource associated with the semaphore (col 2 lines 58-63). It would have been obvious to one of ordinary skill in the art at the time of the invention to try to modify Joffe to teach using a semaphore acknowledge message. One would be motivated by the desire to include some way of indicating the granting of a semaphore.

25. Regarding claim 18, Joffe teaches that when the thread of instructions is in the inactive state, execution of the instructions ceases and the execution circuitry does not

poll the semaphore entity to determine a status of the semaphore request message (col 2 lines 54-58, wherein it is inherent that suspended tasks do perform any functions).

26. Claims 11-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wenniger (US Pat No. 6,018,785) in view of Haghighat et al. (US PG Pub No. US 2005/0080760 A1).

27. Regarding claim 11, Wenniger teaches an apparatus comprising:

an execution circuit to receive and execute a thread of instructions, wherein the execution circuit transmits a semaphore request message and places the thread in an inactive state in response to the thread of instructions requiring a resource having an associated semaphore (col 6 lines 12-22, wherein the requesting process must await an interrupt from the semaphore); and

a semaphore entity coupled with the execution circuit to receive the semaphore request message from the execution circuit and to selectively grant control of the semaphore in response to the semaphore request message by transmitting a semaphore acknowledge message to the execution circuitry, wherein the execution circuitry, in response to receiving the semaphore acknowledge message, removes the thread of instructions from the inactive state and grants the resource to the thread when the resource becomes available (col 6 lines 12-22, where upon receiving the interrupt, the thread queries the semaphore; col 6 lines 40-42, wherein the resource is granted).

28. Wenniger does not explicitly teach that the thread of instructions is placed in an inactive state. Wenniger only teaches that the thread awaits the interrupt from the hardware semaphore. However, it would have been obvious to one of ordinary skill in the art at the time of the invention, that the thread would be placed in an inactive state. It is well known in the art that threads are stalled when the resources that they require are unavailable. One would be motivated by the desire to reduce idle execution time of threads awaiting resources as is well known in the art.

29. Wenniger also does not teach releasing the thread, in response to the thread finishing with the resource, only if the thread is not dependent on any subsequent thread. Haghighat teaches that suspending a thread (i.e. releasing the thread) that owns a lock required by a subsequent thread, can result in a deadlock situation if the resumption of the suspended thread relies on the completion of the subsequent thread whose progress is dependent on the lock of the suspended thread ([0004]). Therefore, it would have been obvious to one of ordinary skill in the art to release the thread in response to the thread finishing with the resource only if there is no thread dependency. One would be motivated by the desire to avoid the deadlock situation taught by Haghighat.

30. Regarding claim 12, Wenniger and Haghighat do not teach further comprising: at least one additional execution circuit to execute threads of instructions; and a thread dispatcher coupled with the execution circuit and at least one additional execution circuit to dispatch threads for execution by selected execution circuits.

31. It would have been obvious to one of ordinary skill in the art, at the time of the invention to add one additional execution circuit to execute threads of instructions and a thread dispatcher. It is well known in the art to add additional execution units to increase processing capability of processors.

32. Claims 19-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joffe et al. (US Pat No. 6,978,330) and Haghighat et al. (US PG Pub No. US 2005/0080760 A1), further in view of Winkeler et al. (US Pat No. 7,237,013).

33. Regarding claim 19, Joffe and Haghighat do not teach placing requests for a semaphore in a queue.

34. However, Winkeler teaches a well known technique of creating a semaphore queue to queue pending requests (col 10 lines 39-47). It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Joffe to teach placing requests for a semaphore in a queue. One would be motivated by the desire to keep track of processes that desire access of the resource by queuing them.

35. Regarding claim 20, Joffe teaches causing a thread to release a semaphore when use of a resource is completed (col 2 lines 62-63).

36. Regarding claim 21, Winkeler teaches automatically granting the resource to the thread whose request is the next request in the queue (col 10 lines 45-47, wherein each component can obtain the lock “in turn”).

Response to Arguments

37. Applicant's arguments with respect to claims 1, 3, 6-7, 11-12, 15-19 and 21 have been considered but are moot in view of the new ground(s) of rejection.

Allowable Subject Matter

38. Claims 13-14 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims including remedying all outstanding 35 USC 112 issues.

Conclusion

39. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

40. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Eric C. Wai whose telephone number is 571-270-1012. The examiner can normally be reached on Mon-Thurs, 9am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng - Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Meng-Ai An/
Supervisory Patent Examiner, Art Unit 2195

/Eric C Wai/
Examiner, Art Unit 2195